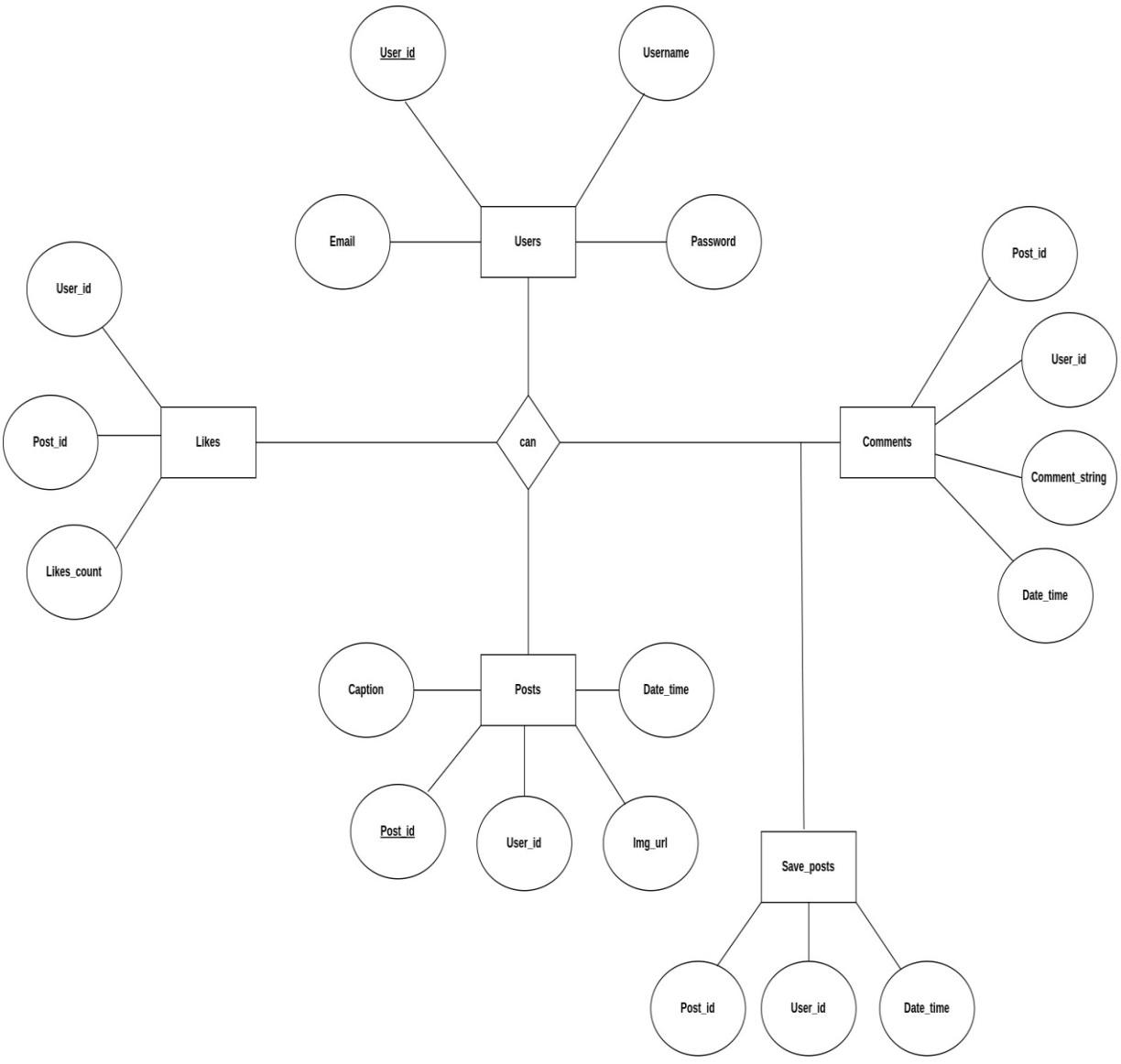
SOCIAL MEDIA MANAGEMENT SYSTEM

INTRODUCTION:-

In today's digital age, social media has become an integral part of our lives, influencing how we connect, communicate, and consume information. With the exponential growth of social networking platforms, managing the vast array of interactions between users and content has become increasingly complex. To streamline this process, we introduce a Social Media Management System, aimed at efficiently handling interactions such as likes, posts, comments, saving posts, and user activities.
The ER diagram serves as a visual representation of the database schema for our Social Media Management System. It illustrates the relationships and interactions between various entities within the system, offering a clear understanding of its structure and functionality.

ER-DIAGRAM:-


User_id
Username
Email
Users
Password
User_id
Post_id
Likes
Likes_count
can
Post_id
User_id
Comment_string
Comments
Date_time
Caption
Posts
Date_time
Post_id
User_id
Img_url
Save_posts
Post_id
User_id
Date_time

RELATIONS:

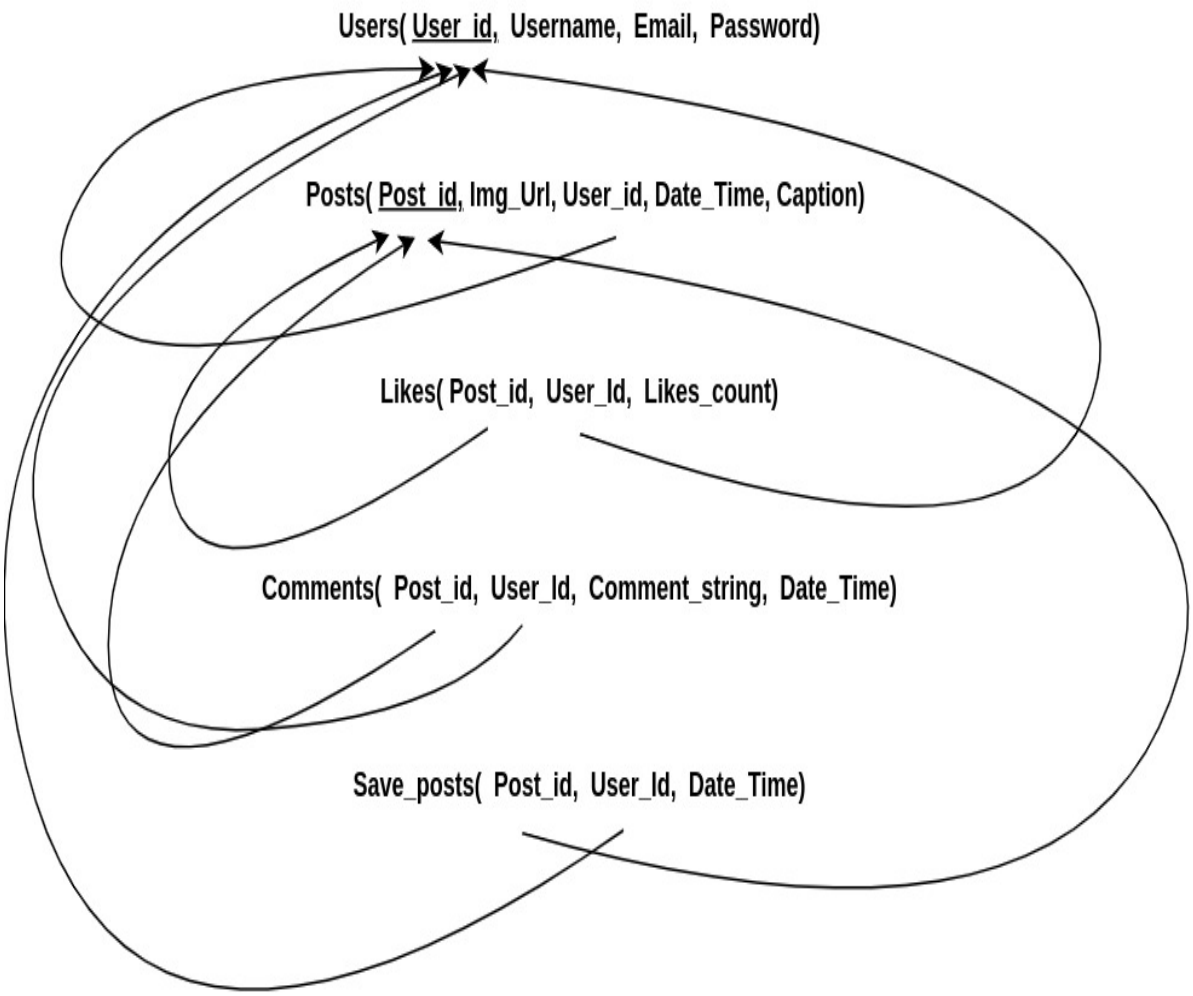
User can **post** images.

User can **like** posts.

User can put **comments** on the posts.

User can save the posts in his/her account.

RELATIONAL SCHEMA:



USERS: this table contains all user information.

CREATE TABLE users (user_id INT, username VARCHAR(255), email VARCHAR(255), PASSWORD VARCHAR(255), PRIMARY KEY(user_id));

INSERT INTO users VALUES(100, 'tanisha_jhaveri', 'tanisha@gmail.com', 't@nisha123');
INSERT INTO users VALUES(101, 'janvi_jain', 'janvi@gmail.com', 'j@nvi123');
INSERT INTO users VALUES(102, 'sahil_doli', 'sahil@gmail.com', 's@hil123');
INSERT INTO users VALUES(103, 'meet_dalal', 'meet@gmail.com', 'meet123');

SELECT * FROM users;

user_id	username	email	password
100	tanisha_jhaveri	tanisha@gmail.com	t@nisha123
101	janvi_jain	janvi@gmail.com	j@nvi123
102	sahil_doli	sahil@gmail.com	s@hil123
103	meet_dalal	meet@gmail.com	meet123

Posts: this table contains all the post which are posted by user.

CREATE TABLE posts(post_id INT PRIMARY KEY, img_url VARCHAR(1000), date_time VARCHAR(200), caption VARCHAR(2000), user_id INT REFERENCES users(user_id));

INSERT INTO posts VALUES(101, 'https://www.google.com/url?sa=i&url=https%3A%2F %2Fhbr.org%2F2020%2F09%2Fdoes-more-money-really-makes-us-morehappy&psig=AOvVaw1O5Drhhiij_UptoEijsWKA&ust=1683729770364000&source=images&cd=vfe&ved=0CBEQjRxqFwoTCMCliaq86P4CFQAAAAAdAAAAABAD', '09-05-2023 10:00pm', 'be happy', '101');

INSERT INTO posts VALUES(102, 'https://www.google.com/url?sa=i&url=https%3A%2F%2Fhbr.org%2F2020%2F09%2Fdoes-more-money-really-makes-us-more-happy&psig=AOvVaw1O5Drhhiij_UptoEijsWKA&ust=1683729770364000&source=images&cd=vfe&ved=0CBEQjRxqFwoTCMCliaq86P4CFQAAAAAdAAAAABAD', '09-05-2023 11:00pm', 'touch the sky with glory', '102');

INSERT INTO posts VALUES(103, 'https://www.google.com/url?sa=i&url=https%3A%2F%2Fhbr.org%2F2020%2F09%2Fdoes-more-money-really-makes-us-more
happy&psig=AOvVaw1O5Drhhiij_UptoEijsWKA&ust=1683729770364000&source=images&cd=vfe&ved=0CBEQjRxqFwoTCMCliaq86P4CFQAAAAAdAAAAABAD', '09-05-2023 11:00pm', 'believe in karma', '103');

SELECT * FROM posts;

post_id	img_url	date_time	caption	user_id
101	https://www.google.com/url?sa=i&url=https%3A%2F%2Fhbr.org%2F2020%2F	09-05-2023 10:00pm	be happy	101
102	https://www.google.com/url?sa=i&url=https%3A%2F%2Fhbr.org%2F2020%2F	09-05-2023 11:00pm	touch the sky with glory	102
103	https://www.google.com/url?sa=i&url=https%3A%2F%2Fhbr.org%2F2020%2F	09-05-2023 11:00pm	believe in karma	103

Likes: this table contains all information about likes made by user on the posts.

CREATE TABLE likes(post_id INT REFERENCES posts(post_id), user_id INT REFERENCES users(user_id), likes_count INT);

INSERT INTO likes VALUES(100, 100, 80);
INSERT INTO likes VALUES(101, 101, 90);
INSERT INTO likes VALUES(102, 103, 100);
INSERT INTO likes VALUES(103, 103, 50);

SELECT * FROM likes;

post_id	user_id	likes_count
101	101	90
102	103	100
103	103	50

Comments: this table contains all information about the comments made by user on the posts.

CREATE TABLE comments(post_id INT REFERENCES posts(post_id), user_id INT REFERENCES users(user_id), comment_string VARCHAR(2000), date_time VARCHAR(2000));

INSERT INTO comments VALUES(100, 100, 'yes smile always', '11-05-2023 9:00pm');
INSERT INTO comments VALUES(101, 101, 'this picture is beautiful', '11-05-2023 10:00pm');
INSERT INTO comments VALUES(102, 102, 'this picture is heart touching', '11-05-2023 11:00pm');

SELECT * FROM comments;

post_id	user_id	comment_string	date_time
101	101	this picture is beautiful	11-05-2023 10:00pm
102	102	this picture is heart touching	11-05-2023 11:00pm

Save_posts:- this table contains information about the posts saved by user in his/her profile.

CREATE TABLE save_posts(post_id INT REFERENCES posts(post_id), user_id INT REFERENCES users(user_id), date_time VARCHAR(200));

INSERT INTO save_posts VALUES(100, 100, '23-10-2023 8:00pm');
INSERT INTO save_posts VALUES(101, 101, '22-10-2023 7:00pm');
INSERT INTO save_posts VALUES(103, 103, '20-10-2023 5:00pm');
INSERT INTO save_posts VALUES(102, 102, '22-10-2023 6:00pm');

SELECT * FROM save_posts;

post_id	user_id	date_time
100	100	23-10-2023 8:00pm
101	101	22-10-2023 7:00pm
103	103	20-10-2023 5:00pm
102	102	22-10-2023 6:00pm

Procedure for getting posts with comments, likes count by just entering name of the user.

```
DELIMITER $$

CREATE
PROCEDURE `s`.`seeUserPosts`(IN uName VARCHAR(20))
	BEGIN
		DECLARE uId INT;
		SELECT user_id INTO uId FROM users WHERE username = uName;
```

```
                SELECT img_url,likes_count,comment_string FROM posts p JOIN
likes l         ON p.post_id = l.post_id JOIN comments c ON p.post_id = c.post_id
                    WHERE p.user_id =   uId;
        END$$

DELIMITER ;
```

CALL `seeUserPosts`('tanisha_jhaveri');

img_url	likes_count	comment_string
https://www.google.com/url?sa=i&url=https%3A%2F%2Fhbr.org%2F2020%2F	90	this picture is beautiful

Trigger for the functionality that if user is deleted then the posts, likes, comments associated with him/her will be automatically deleted

```
DELIMITER $$

CREATE
 TRIGGER `userDel` AFTER DELETE ON `users`
FOR EACH ROW BEGIN
        DELETE FROM posts WHERE user_id = old.user_id;
        DELETE FROM likes WHERE user_id = old.user_id;
        DELETE FROM comments WHERE user_id = old.user_id;
END;
$$

DELIMITER ;
```

user_id	username	email	password
100	tanisha_jhaveri	tanisha@gmail.com	t@nisha123
101	janvi_jain	janvi@gmail.com	j@nvi123
102	sahil_doli	sahil@gmail.com	s@hil123
103	meet_dalal	meet@gmail.com	meet123

DELETE FROM users WHERE user_id = 101;

user_id	username	email	password
100	tanisha_jhaveri	tanisha@gmail.com	t@nisha123
102	sahil_doli	sahil@gmail.com	s@hil123
103	meet_dalal	meet@gmail.com	meet123

post_id	img_url	date_time	caption	user_id
102	https://www.google.com/url?sa=i&url=https%3A%2F%2Fhbr.org%2F2020%2F	09-05-2023 11:00pm	touch the sky with glory	102
103	https://www.google.com/url?sa=i&url=https%3A%2F%2Fhbr.org%2F2020%2F	09-05-2023 11:00pm	believe in karma	103

post_id	user_id	likes_count
102	103	100
103	103	50

post_id	user_id	comment_string	date_time
102	102	this picture is heart touching	11-05-2023 11:00pm